> activity before starting

1. meet someone
   a. find someone you don’t know who isn’t talking with anyone else
   b. introduce yourself
   c. discuss: what’s the difference between a $programmer and a $software engineer?

> what lies beyond the code?

@AgileVietnam
@AgilistAndre
@Mindvalley
#WhatLiesBeyondCode

Café
HONDA
110

YẾN SÀO SÀI GÒN ANPHA
QUÀ TẶNG CHO SỨC KHỎE
THƯƠNG HIỆU ĐÃ ĐƯỢC BẢO HỘ ĐỘC QUYỀN
YẾN SÀO SÀI GÒN ANPHA
THƯƠNG HIỆU ĐÃ ĐƯỢC BẢO HỘ ĐỘC QUYỀN
YẾN SÀO SÀI GÒN
Yến Sào THĂNG LONG
MILANO
MILANO
CAFE
24
24
VINASUN TAXI
VINASUN TAXI
VINASUN TAXI
HONDA
HONDA

SURF

Việt Nam

Trang
Nữ
DUYTRI
Wave

LICH THI ĐẤU VÒNG CHUNG KẾT
EURO 2012
TAXI TẢI
Gọi đâu có đó
Hóa Thanh
ĐT: 0979.204.084
0925.05 05 04
0905.61 87 19
Vị Đậm Đà
0925.05
0905.61 87 19
ĐT:09356.23 456
ĐT:09356.23
CÀ PHÊ CAM
CẮT TÓC
Nam-
THUỲ VÂ
VERSACE
Stop Making Sense

> activity before starting

1. poll
   a. go to https://www.menti.com and input the code:
      - 68 08 80

   b. answer the poll question: What is your role?

LiveSlides web content

To view

Download the add-in.

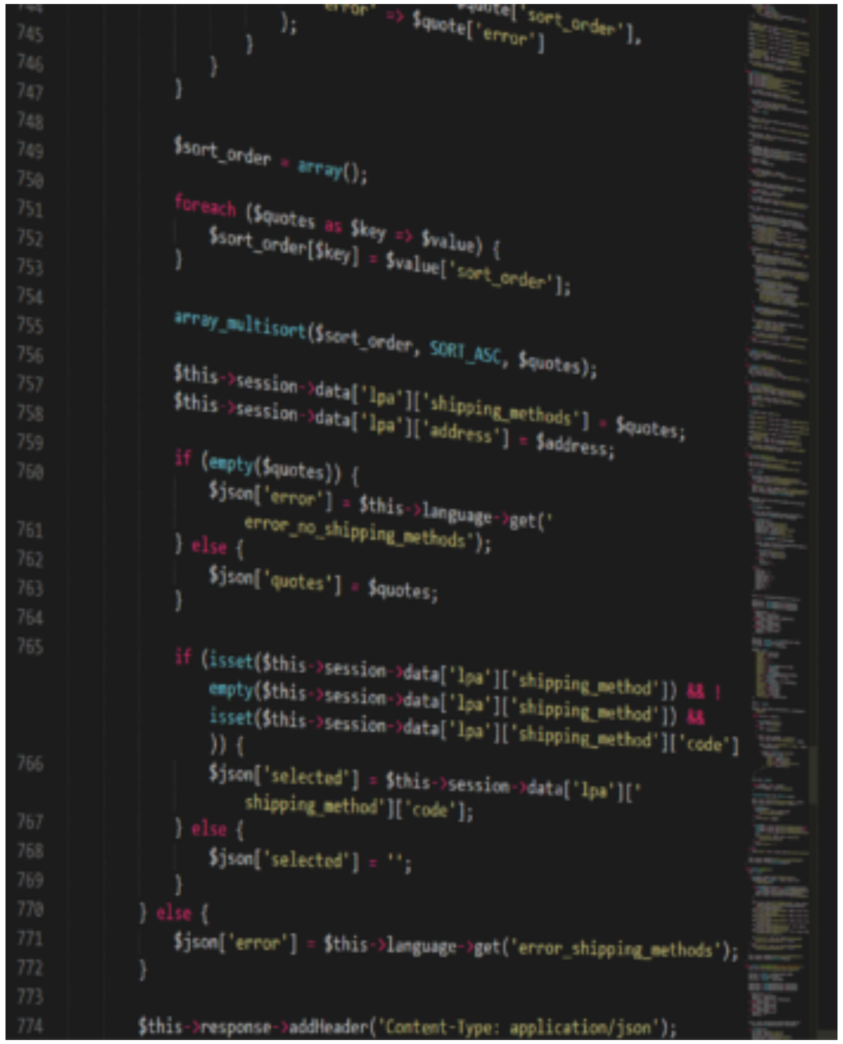
liveslides.com/download

Start the presentation.

楚河
漢界

> what is
check-mate in
software development?

> check-mate in
software development
is to
add value to users



```
                        'error' => $quote['error']
                    );
                }
            }
        }

        $sort_order = array();

        foreach ($quotes as $key => $value) {
            $sort_order[$key] = $value['sort_order'];
        }

        array_multisort($sort_order, SORT_ASC, $quotes);

        $this->session->data['lpa']['shipping_methods'] = $quotes;
        $this->session->data['lpa']['address'] = $address;
        if (empty($quotes)) {
            $json['error'] = $this->language->get('
                error_no_shipping_methods');
        } else {
            $json['quotes'] = $quotes;
        }

        if (isset($this->session->data['lpa']['shipping_method']) && !
            empty($this->session->data['lpa']['shipping_method']) &&
            isset($this->session->data['lpa']['shipping_method']['code']
            )) {
            $json['selected'] = $this->session->data['lpa']['
                shipping_method']['code'];
        } else {
            $json['selected'] = '';
        }
    } else {
        $json['error'] = $this->language->get('error_shipping_methods');
    }

    $this->response->addHeader('Content-Type: application/json');
```

?

Code

Test

Deploy

Merge

Code Review

Build

bac

nning

Team

t

offee

Inte

y Huddle

Set up

Onboard new team mates

[✓ Code]

Meetings

Test

Merge

Deploy

Code Review

Have pizza

Play

Build

backlog

Research

Choose

Sprint Planning

Learn

Discuss architecture

Compare

Team Building

Inspect and Adapt

Talk to Customer

Support

Production

Have coffee

Interview candidates

Purchase

Integrate

Daily Huddle

Set up CI/CD

Onboard new team mates

Code

“The fastest code is the code which does not run."

"The code easiest to maintain is the code that was never written.”

-Robert Galanakis

Code

> “The best code is no code at all. Every new line of code you willingly bring into the world is code that has to be debugged, code that has to be read and understood, code that has to be supported. Every time you write new code, you should do so reluctantly, under duress, because you completely exhausted all your other options.”

-Jeff Atwood

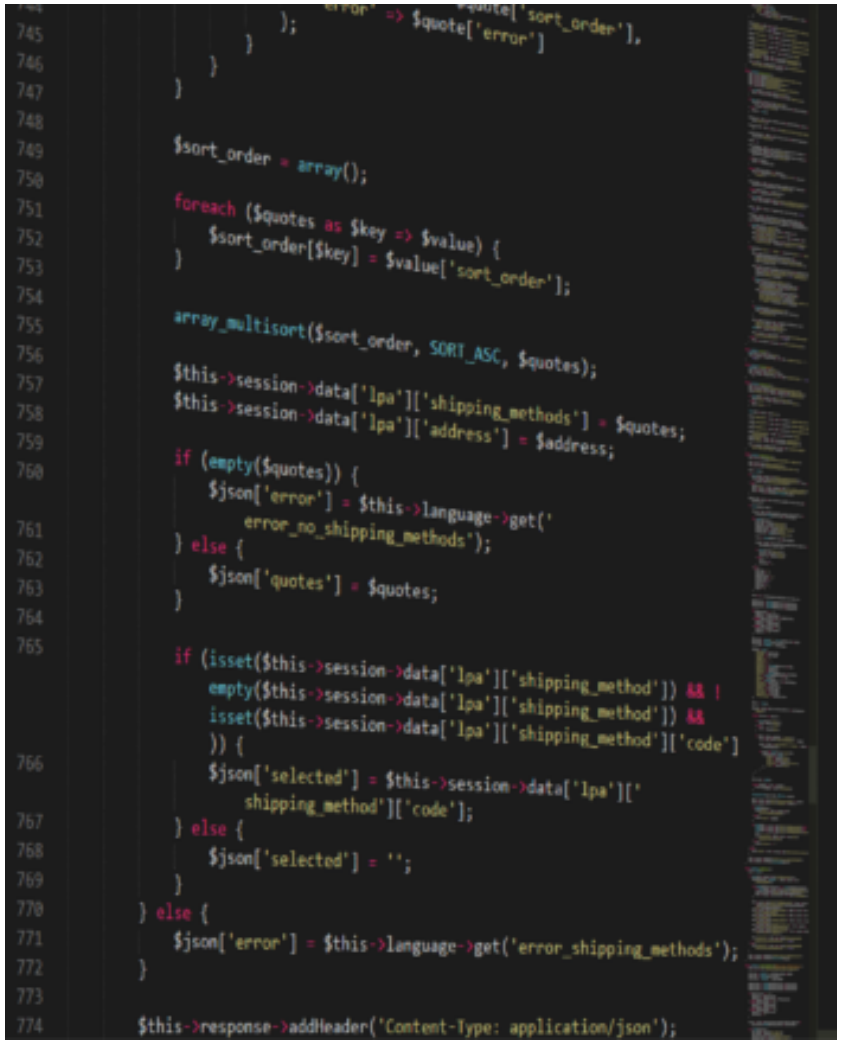
Code

“Code is bad. It rots. It requires periodic maintenance. It has bugs that need to be found. New features mean old code has to be adapted. The more code you have, the more places there are for bugs to hide. The longer checkouts or compiles take. The longer it takes a new employee to make sense of your system. If you have to refactor there's more stuff to move around."

-Rich Skrenta, 2007

“Code is produced by engineers. To make more code requires more engineers. Engineers have n^2 communication costs, and all that code they add to the system, while expanding its capability, also increases a whole basket of costs. You should do whatever possible to increase the productivity of individual programmers in terms of the expressive power of the code they write. Less code to do the same thing (and possibly better). Less programmers to hire. Less organizational communication costs.”

-Rich Skrenta, 2007



```
                        'sort_order' => $quote['sort_order'],
                        'error'      => $quote['error']
                    );
                }
            }
        }

        $sort_order = array();

        foreach ($quotes as $key => $value) {
            $sort_order[$key] = $value['sort_order'];
        }

        array_multisort($sort_order, SORT_ASC, $quotes);

        $this->session->data['lpa']['shipping_methods'] = $quotes;
        $this->session->data['lpa']['address'] = $address;
        if (empty($quotes)) {
            $json['error'] = $this->language->get('
                error_no_shipping_methods');
        } else {
            $json['quotes'] = $quotes;
        }

        if (isset($this->session->data['lpa']['shipping_method']) && !
            empty($this->session->data['lpa']['shipping_method']) &&
            isset($this->session->data['lpa']['shipping_method']['code']
            )) {
            $json['selected'] = $this->session->data['lpa']['
                shipping_method']['code'];
        } else {
            $json['selected'] = '';
        }
    } else {
        $json['error'] = $this->language->get('error_shipping_methods');
    }

    $this->response->addHeader('Content-Type: application/json');
```

```
> deliver value to the user
```

> what’s the difference between a $programmer and a $software engineer?

```
> delivering value to the user

> how (fast) we deliver value?

    > $efficiency

> what is value?

    > $effectiveness
```

> delivering value to the user

- > Delivering things right
  - > $efficiency
- > Delivering the right things
  - > $effectiveness

KELLY
HEINZ 57
HEINZ
TOMA
KETC
JAIM
PORTLAN

> assessing value

> delivering value to the user

> delivering the right things

> > $effectiveness

```
> three levels of maturity
  of software developers

> $programmer

> $software engineer

> $software intrapreneur
> what lies beyond the code?

          > value
```

> what lies beyond the code?

> key take-aways:

> $efficiency -> improving the whole value stream, not just the speed of coding

> $effectiveness -> building the right thing

> what lies beyond the code?

> action items (for Monday):

- > find 1 way to improve $efficiency

- > find 1 way to improve $effectiveness